AFRL-VS-PS-
TR-2005-1061

AFRL-VS-PS-
TR-2005-1061

ASYNCHRONOUS ARCHITECTURES FOR LARGE-INTEGER PROCESSORS WITH APPLICATIONS TO SECURITY

Alain J. Martin

Situs Logic
1442 Lomita Drive
Pasadena, CA 91106

15 April 2005

Final Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776

AFRL-VS-PS-TR-2005-1061

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data, does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report has been reviewed by the Public Affairs Office and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

If you change your address, wish to be removed from this mailing list, or your organization no longer employs the addressee, please notify AFRL/VSSE, 3550 Aberdeen Ave SE, Kirtland AFB, NM 87117-5776.

Do not return copies of this report unless contractual obligations or notice on a specific document requires its return.

This report has been approved for publication.

//signed//

JAMIE BASTIDAS, JR., Capt, USAF
Project Manager

//signed//

JOHN P. BEAUCHEMIN, Lt Col, USAF
Acting Chief, Spacecraft Technology Division
Space Vehicles Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
15/04/2005

2. REPORT TYPE
Final Report

3. DATES COVERED *(From - To)*
15/04/2004 to 15/04/2005

4. TITLE AND SUBTITLE
Asynchronous Architectures for Large-Integer Processors with Applications to Security

5a. CONTRACT NUMBER
FA9453-04-M-0112

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
65502F

6. AUTHOR(S)
Alain J. Martin

5d. PROJECT NUMBER
3005

5e. TASK NUMBER
VP

5f. WORK UNIT NUMBER
HV

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Situs Logic
1442 Lomita Drive
Pasadena, CA 91106

8. PERFORMING ORGANIZATION REPORT NUMBER
Situs-TR-05-02

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Space Vehicles Directorate
3550 Aberdeen Ave., SE
Kirtland AFB, NM 87117-5776

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/VSSE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-VS-PS-TR-2005-1061

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The objective of this research was to develop new architectures for cryptographic hardware that offer extremely high throughput, algorithm flexibility and system upgradability, radiation hardness by design, and low power. The feasibility of an asynchronous (clockless) architecture combining a dedicated large integer processor, an FPGA, and a simple processor was investigated. A novel approach to making asynchronous circuits SEU-tolerant was developed and simulated. Simulation and analysis demonstrate that such an architecture combines high throughput, adaptability, and excellent resistance to SEUs. The systems envisioned would be built out of quasi-independent components that can be commercialized stand-alone and in different configurations, enhancing the upgradability and flexibility of the product range.

15. SUBJECT TERMS Cryptography, RSA, ECC, NTRU, AES, DES, asynchronous, clockless, radhard-by-design, SEU-tolerant, low-power, FPGA, large-integer processor, Montgomery multiplication

16. SECURITY CLASSIFICATION OF:

a. REPORT
Unclassified

b. ABSTRACT
Unclassified

c. THIS PAGE
Unclassified

17. LIMITATION OF ABSTRACT
Unlimited

18. NUMBER OF PAGES
14

19a. NAME OF RESPONSIBLE PERSON
Capt Jamie Bastidas

19b. TELEPHONE NUMBER *(include area code)*
(505) 853-2202

1 Summary

This Phase I study was concerned with the design and implementation of asynchronous architectures for a large integer processor (LIP) able to perform arithmetic operations, in particular multiplications, on integers of a size that current architectures cannot handle efficiently (on the order of 5000 bits long). The main applications are in the area of security, in particular public-key and mixed private-key/public-key cryptography. It was concluded that a LIP-based system would be able to perform public-key (RSA or ECC) encryption at a fraction of the runtime and/or energy consumption of current hardware platforms. The LIP would be offered in radiation-hardened-by-design (RHBD) and non-hardened versions to suit a range of applications.

The LIP computational engine proposed was to be software reconfigurable for different cryptographic protocols and also for the different phases in each protocol. The LIP engine consists of a large-integer datapath with relatively standard units (adder, multiplier, etc.) but with data formats specifically chosen to permit efficient implementation of larger operations. The design uses hardware word sizes of between 128 and 512 bits in order to implement integer operations on 1024-bit or larger operands. This part of the design mainly addresses public-key cryptosystems such as RSA.

Furthermore, the architecture would include finer-grain reconfigurable hardware (such as that found in a normal FPGA) in order to accelerate certain cryptographic protocols (especially those including private-key cryptography). This part of the design mainly addresses private-key cryptosystems such as triple-DES, AES, and many classified systems.

The conclusion of this study is twofold. First, the combination of a large-integer datapath and a reconfigurable private-key cryptography system would make the LIP a true "cryptosystem on a chip," capable of all the computations necessary for key management, digital signatures, and encryption/decryption. Second, the combination of asynchronous techniques and tolerance to single-event-upsets would give the proposed architectures enormous advantages in terms of energy efficiency and robustness.

2 Introduction

The need for secure communication in military and homeland-security related computer systems is obvious—from satellite communications to battlefield wireless tactical networks and sensor networks. In the civilian domain, the use of cryptography is expanding rapidly from server protocols to smartcard transactions, including commercial satellites, electronic financial transactions, medical information exchanges, digital signatures, handheld and portable devices, etc.

As a result, it can be stated that, in the near future, all computer systems will include or access cryptographic functions. By their very nature, cryptographic functions rely on computationally-demanding algorithms, the implementation of which often strains the system in which they are embedded, in terms of cost or performance. Because of the variety of cryptographic algorithms currently in use (e.g., RSA, ECC, NTRU, AES, DES), and the need to constantly upgrade the key sizes (e.g., from 128 bits to 192 to 256), another requirement placed on those architectures is that they be flexible and easily upgradable. Finally, space and military applications also require that the hardware involved be energy efficient and radiation resistant.

Applications in new areas like wireless and adhoc networks are implemented on lightweight hardware for which cryptographic operations are impossible to perform in reasonable time and with a reasonable amount of energy. With the rapid increase of embedded network uses in home, commercial, industrial, automotive, and defense applications, the lack of good security solutions for those networked systems is particularly critical.

The basis for secure communication is message encryption and decryption. Traditionally, encryption and decryption have been done using private keys: both sender and receiver of a message share a secret key that they use to encrypt and decrypt the message (like in the well-known DES and AES protocols). Unfortunately, these protocols require the sharing, and therefore often the exchange, of secret information, which makes them vulnerable. Great progress in the area was made with the invention of public-key cryptography and in particular the RSA algorithm. Those protocols require the use of a private (secret) key and a public key. Anybody wishing to send a message to Alice encrypts it with Alice's public key. Only Alice can decrypt the message with her private key. A way to break the protocol (it is believed, but not proved, that it is the only way) would be for an attacker to factor the public key into its prime factors.

Therefore, the security of those algorithms require the use of integer keys large enough to make their factorization impractical in the state-of-the-art technology. As a consequence, the operations required on those integers, in particular multiplications, are costly in terms of runtime and energy consumption. Currently, private-key protocols like AES use public-key techniques to communicate the private key among users. In such protocols, the public-key part is usually the weakest link (see section 3.2).

The purpose of this project was to make robust public-key cryptography available for those applications by designing an asynchronous large-integer processor (LIP) that, it is believed, can perform public-key encryption (or the basic arithmetic operations used by such a protocol) using large keys at a fraction of the cost in runtime and energy consumption compared to state-of-the-art synchronous (clocked) systems. Reconfigurable hardware would be used in order to speed up private-key encryptions and other operations.

3 Method and Procedures

3.1 Radhard Advantages of Asynchronous Logic

Asynchronous VLSI adds other advantages to a LIP used for cryptography. Asynchronous circuits can be energy-efficient and robust to variations in temperature and voltage supply, and recent work by Situs has shown that asynchronous circuits can be made radiation hard by design without too much difficulty. Those combined qualities are particularly interesting for space applications.

The recent work by Situs on radiation hardening by design for asynchronous circuits has built on the observation that (delay insensitive or quasi delay-insensitive) asynchronous circuits are tolerant to delay variations, which makes them naturally tolerant to certain radiation effects. The radiation effects that result in timing variations are accumulated dose effects, dose-rate effects, and certain kinds of single-event effects (SEEs). Situs has augmented the natural tolerance of the asynchronous circuits to these effects with redundancy schemes to handle single-event upset (SEU). The techniques are comparable in penalty to triple-modular redundancy (TMR) techniques used in synchronous systems but allow the asynchronous advantages of correctness independent of timing to be maintained.

3.2 Advanced Encryption Standard and Public-Key Encryption

The Advanced Encryption Standard (AES) was introduced by the National Institute of Standards and Technology (NIST) as Federal Information Processing Standard (FIPS) 197. The basic version of AES uses 128-bit keys, and it is applicable to sensitive (unclassified) government data. The National Security Agency's CNSS Secretariat has recently (June 2003) reviewed AES and found it to be adequate to protect classified information up to and including the SECRET level; the basic 128-bit version was not considered adequate for TOP SECRET data—for this, NSA requires the use of 192- or 256-bit keys.

While the use of AES on its own may have been sufficient for protecting most government data in the past, using AES efficiently in highly distributed systems such as sensor networks is very difficult, and it is likewise impossible to use AES on its own in many commercial settings. The problem is that AES is a private-key (symmetric) cryptosystem, and while distributing keys might be a viable alternative in a traditional military setting, it is much more difficult in highly distributed systems where it may be impractical or impossible to bring all the parties together to agree on AES keys before they are deployed in the insecure environment. In many commercial applications, users are entering and leaving the system at a high rate (e.g., they may be initiating sessions with a web server in e-commerce applications), and each new user needs a secure method of communicating with the server.

The most practical solution to distributing keys in situations when the communicating parties cannot be brought together in a secure environment beforehand is to use some sort of public-key protocol to distribute the keys and then continue communicating using a private-key system. Private-key systems such as AES are much more computationally efficient than public-key systems, but if the messages to be sent are very short (e.g., in sensor networks), it may make sense to forgo the private-key system entirely. Public-key protocols are also necessary if the transmissions are to be digitally signed. (See FIPS 186 and FIPS 196.)

Guidelines for key distribution and maintenance (key management) are under development at NIST. Drafts of the proposed FIPS have been in circulation, and the main concerns of the NIST expert group are clear. One of the most important points is the "weakest link" observation: a cryptographic protocol is only as strong as its weakest step. For instance, a system using AES-128 for encrypting bulk data cannot be considered "128-bit strong" if the private keys are distributed using a public-key protocol that is weaker than AES-128. (In this context, "weaker" means that less computing power, fewer CPU-hours or CPU-years, is necessary to break the public-key protocol than to break AES-128.)

3.3 The Need for Increased Processing Power

Most current cryptosystems that use RSA use 1024-bit keys; in fact, 1024-bit RSA has become the de facto standard for key exchange in commercial Internet traffic. Numerous commercial vendors supply accelerator chips that can be used to speed up 1024-bit RSA in commercial environments.

The difficulty of attacking RSA is entirely based on the difficulty of factoring large numbers (the keys) into their prime factors. There has been a great deal of progress in this field in the past decade, and 1024-bit integers have become orders of magnitude easier to factor. Furthermore, improved computer performance and lower costs have made it yet orders of magnitude easier to factor large numbers.

Taking recent advances into account, the NIST group has determined that 1024-bit RSA is only about as secure as 80-bit symmetric encryption. So if a system uses 1024-bit RSA to set up the keys for an AES-128 session, the RSA step is the weak link. In order to match the security of AES-128, one would have to use 3072-bit RSA and in order to match the security of the protocols NSA has suggested for classified information, 7680-bit RSA or 15,360-bit RSA would be necessary. Microsoft has also shown distrust in 1024-bit RSA by selecting 2048-bit RSA for performing various security-related operations on their Xbox game console.

It seems clear that RSA keys are going to get larger in the near future, and this is of great importance to hardware manufacturers. Today's hardware RSA accelerators will simply not be adequate for accelerating calculations with the very large keys that will become commonplace. This is because the difficulty of performing RSA operations (normal operations, not attacking the code) is roughly *cubic* in they key length: i.e., 2048-bit RSA operations take eight times as long to perform as 1024-bit operations; 15,360-bit operations take 3,375 times as long!

The LIP design was intended to be a computer with a specialized datapath operating on medium-sized integers (256–512 bits, still much larger than any normal microprocessor).

4 Results and Discussion

4.1 Why an Asynchronous Architecture?

Fast multiplication of large integers requires hardware solutions that are physically large, typically a large array of cells. Currently, multiplication circuitry already occupies the largest area of a processor chip, sometimes up to 50%. An integer multiplier of the type envisioned requires a chip area beyond what can be reliably designed in a conventional technology. Large chip areas create parameter gradients that are difficult to control in a clocked technology. Particularly, clock distribution on such a large chip creates a clock skew that can be corrected only at the price of unacceptable losses in performance. Other parameter variations across the chip, for instance in temperature or threshold voltage may also result in timing variations that negatively affect the clock period resulting either in malfunctions or in decreased performance. Finally, the power consumption of such a chip would be prohibitively high in conventional technology unless aggressive clock gating is used, which instead necessitates extreme expenditures of time and effort in the design process.

Asynchronous VLSI circuits do not use a clock. The type of asynchronous circuits developed by Situs are called quasi delay-insensitive (QDI) because they make minimal assumptions on delays in operators and wires. Because they are largely delay insensitive, those circuits are very robust to variations in physical parameters: voltage, temperature, wire lengths, threshold voltages, doping, etc. Furthermore, all issues related to clock distribution are eliminated, and communications between modules are totally insensitive to delays. It is therefore very easy to array large ensembles of asynchronous modules without penalty.

The PI's DARPA-sponsored research in energy-efficient asynchronous architectures has demonstrated that asynchronous circuits can be very energy-efficient: (1) because such circuits do not use clocks, a large fraction of the energy spent in the clock is saved; (2) when an asynchronous component is not use, it does not consume any energy, essentially providing the equivalent to perfect clock gating; (3) because it is easy to vary the supply voltage over a wide range, it is easy to trade speed for low energy in an asynchronous circuits; and (4) asynchronous circuits are glitch-free by design, eliminating the considerable power lost to glitching transitions. For ultra-low-power applications such as encryption in "smart dust" sensor networks, it may be necessary to operate in or near subthreshold. The operation of an asynchronous microprocessor at 0.4 volts supply voltage has previously been demonstrated, in a 1.6-μm CMOS process with a nominal threshold voltage of between 0.9 and 1.0 volts, and it is believed such such low-voltage operation can be translated into practically usable ultra-low-power operation.

4.2 Proposed Architecture

The envisioned LIP architecture consists of a large array of computing elements communicating by message passing. The architecture would mix granularities (large grains such as large-wordsize multipliers and small grains such as FPGA LUTs) in order to match the needs of different cryptographic protocols and the different phases within each cryptographic protocol. FPGAs have been shown to be helpful for implementing private-key systems such as AES, and it is also true that FPGAs can be used to configure partially evaluated arithmetic functions for certain public-key protocols.

Determining the proper level of granularity and mixing of different granularities was one of the most important objectives of this Phase I study.

Four different designs, at different performance and cost levels, have been studied. The designs have datapaths with sizes ranging from 128 bits to 512 bits and include radiation-hardened and non-hardened designs.

In 0.13-μm CMOS technology, a non-hardened 256-bit LIP would be able to perform 46K (1024-bit) RSA decryptions per second using 10 watts of power; a 512-bit version would do 184K RSA decryptions while using 40 watts. These numbers are computed for operation at the nominal voltage; by using voltage scaling, these designs could perform tens to hundreds of RSA decryptions per second using power in the microwatt to low milliwatt range. The nearest commercial competitor is the Broadcom BCM5821, which manages 4K 1024-bit RSA decryptions per second at 3 watts. The private-key part of the LIP would be capable of 50 Gb/second at 1 watt or 100 Gb/second at 2 watts. The choice of design would depend on cost issues: the 512-bit design would occupy approximately 150 mm^2 in 0.13-μm technology (a large chip), whereas the 256-bit design would occpy approximately 50 mm^2. Radiation-hardened-by-design versions would be larger and slower (approximately three times the area and 2/3 the speed).

The LIP's public-key cryptographic operations are all performed in the Large Integer Datapath (LID). The LID would be optimized for the following two goals:

- Maximum throughput
- Support of different key sizes

Since each operation requires a very large number of bit-ops, it is important to apply a number of arithmetic algorithms in order to reduce the number of bit-ops to close to what is achieved by the best known algorithms. A particular combination of arithmetic algorithms (Chinese Remainder Theorem, the Montgomery modular multiplication algorithm, higher-radix exponentiation, and Karatsuba multiplication) was selected in the reference LIP software.

Because each algorithm offers a 1.5X–4X factor of bit-op reduction, and these factors multiply, it is important to apply as many of these algorithms as possible *simultaneously*. The best way to allow all algorithms to be simultaneously implemented while supporting different key sizes is to implement all algorithms in software. This approach allows the use of a simple hardware abstraction with simple operation semantics, which software can exploit efficiently, and which hardware can implement efficiently.

Most of the bit-ops in the selected algorithms occur in integer multiply operations. Therefore, to execute these bit-ops efficiently, the LID uses the simplest possible scalable abstraction, namely integer multiplication of a fixed word size. Also, as the throughput of the LID is limited by the throughput of the large integer multiplier, the large integer multiplier is a pipelined non-iterative multiplier, which gives the maximum possible throughput. This approach avoids the following pitfalls which are common in academic RSA implementations:

- Building some of the arithmetic algorithms into hardware, and ending up with a complicated hardware abstraction that makes it difficult to implement the remaining arithmetic algorithms in software.
- Using an iterative multiplier, which reduces throughput. While the multiplier area is smaller, the register area (needed to implement the crucial software algorithms) is not smaller, so the lost throughput cannot be recaptured by replicating the design until its area matches that of a design using a non-iterative multiplier.

For example, one design which claims to be the best design of this type in 2004, achieves an overall performance on 1024-bit RSA decryption of just 368 keys per second, which is slower than software running on a Broadcom 5821 (4,000 keys per second) or Alpha processor (437 keys per second when scaled to 600 MHz 21164).

The LIP avoids this problem by using a high-throughput non-iterative multiplier with a simple programming abstraction.

The proposed LIP architecture would use very large amounts of hardware for integer calculations compared to, say, a modern RISC or x86 CPU (hundreds of times as much).

The vast amount of hardware means that the LIP can operate at relatively high power levels, which may be acceptable in a server application, but would seem impractical in an embedded application, where only a few milliwatts might be available for cryptographic activities. The mismatch is illusory. In order to perform a given operation (e.g., a 1024-bit RSA decryption) in a given amount of time with as little energy as possible, one should use *a large amount of hardware, where each piece operates very slowly, but the whole operates at the desired speed.*[1]

One may consider a hypothetical "simple-LIP" based on a 512-bit array multiplier. This is a simple architecture, and using relatively straightforward algorithms, such a device can be expected to perform between 135,000 and 150,000 1024-bit RSA decryptions[2] per second while consuming 40 watts of power at a supply voltage of 1.0 volts in 0.13-μm CMOS; the circuits operate at about 500 MHz in this machine. For this calculation, the use of the Chinese Remainder Theorem for the representation of numbers has been assumed. These performance figures could be compared to around 250 RSA decryptions per second for a 2.1 GHz Intel Pentium 4 processor, whose power consumption and implementation technology are similar. As a second comparison point, Broadcom's fastest current encryption chips (the BCM5821 Super E-Commerce Processor) can do only 4,000 1024-bit RSA operations per second; this chip is also only targeted at this particular performance point, unlike the LIP, which will be able to operate over a wide range of speeds and power levels.

Seventy-five watts' power consumption is of course out of the question in an embedded application; the very same processor could be operated at a supply voltage of 0.2 volts, in which case it would consume 10 microjoules of energy per RSA decryption ($\frac{1}{25}$ as much as at 1 volt). In an embedded system, this would permit tens to hundreds of RSA decryptions per second while using only microwatts or milliwatts of power.

5 Conclusion

A conclusion of this study is that the actual LIP implementation would perform better than the numbers in this proposal, mainly owing to improved algorithms. Schemes for using squarers (rather than multipliers) and various schemes for fast modulus calculations have been considered that would lead to both energy and speed improvements. Further improvements can be expected from future work in design techniques for fast squarers and multipliers, since the large wordsizes used in the LIP permit a wider range of hardware implementations than has formerly been studied.

It would be impossible to achieve the performance figures of the LIP, at either end of the speed scale, using small wordsizes. At the upper end of the scale, the amount of hardware simply would not be sufficient to perform at the rate of the LIP; at the lower end, performance would suffer so much that it would not be possible to lower the supply voltage to 0.2 volts—an architecture with a smaller wordsize would have to operate at a higher voltage and therefore spend more energy to perform the task. In this sense, the proposed architecture permits trading speed for energy over a wider range than other architectures.

[1] Given control over the threshold voltage (either through fabrication or through back biasing), this conclusion extends even into the regime where leakage currents dominate.

[2] 2048-bit decryptions can be assumed to run one eighth as fast in all cases.

The speed is limited entirely by power and heat-dissipation concerns: operated at full speed, the simple-LIP could perform 300,000 RSA decryptions per second using around 300 watts of power—this speed could be practically achievable for short periods of time but is out of the question for sustained operation unless extreme measures are taken for cooling and power distribution.

The low-energy operation of the LIP is only practical with an asynchronous implementation. First, unless the LIP is used continuously (unlikely in many embedded applications), clock gating will be absolutely necessary for energy efficiency; to work properly, such clock gating would have to be very fine-grained (as the multiplier is many pipeline stages deep) and would be difficult to implement. Secondly, at low supply voltages in CMOS processes, delays are very difficult to predict and can be quite variable. The delays are affected by things like temperature and device aging; as one operates a device at voltages near or below the digital threshold voltage of a process, temperature and device aging have exponential effects on delays. It would be difficult or impossible to account for these effects in a traditional synchronous implementation. Experimental evidence has shown that this is true: even carefully designed synchronous chips are not able to trade speed for energy as efficiently as asynchronous ones.

A final advantage to the asynchronous LIP implementation is realized by the use of the low-latency pipeline stages pioneered at Caltech in the MiniMIPS and other projects. Pipelined array multipliers such as the ones proposed in the LIP generally use "carry-save" adders, which adds to computation latency in synchronous implementations (the data has to wait for the clock). In an asynchronous implementation, a carry-save adder can be implemented with much less latency because there is no need to slow down the data in order to match the clock. Also, low-energy, simple ripple-carry adders can be used in place of more complicated carry-lookahead adders without compromising performance.

Situs still strongly believes in the soundness and advantages of the proposed architecture, and hopes to pursue the project.

DISTRIBUTION LIST

DTIC/OCP
8725 John J. Kingman Rd, Suite 0944
Ft Belvoir, VA 22060-6218 1 cy

AFRL/VSIL
Kirtland AFB, NM 87117-5776 1 cy

AFRL/VSIH
Kirtland AFB, NM 87117-5776 1 cy

Situs Logic
1442 Lomita Drive
Pasadena, CA 91106 1 cy

Official Record Copy
AFRL/VSSE, Capt Jamie Bastidas 1 cy